LEARNING SPEECH STRUCTURE: A DYNAMICAL SYSTEM PERSPECTIVE

A. Ramos-Cabello, J. Quiñonero-Candela, A. Gallardo-Antolín and F. Díaz-de-María
Dept. of Communication Technologies
EPS-Universidad Carlos III de Madrid, Spain

Abstract.- In this paper we present an alternative speech prediction system. The new approach consists in predicting from a vector of non-consecutive samples, being *T* the distance between samples. The dimension of the vector *d* and the distance *T* have been estimated following the algorithms described in [Abarbanel 98]. The results prove that the suggested system achieves similar results to that of a conventional one but with a computational load much lower. Furthermore, *T* can be selected in the neighbourhood of its optimal value without loss of performance.

I. INTRODUCTION

Linear prediction has been one of most preponderant tools in the field of speech processing during the last three decades; nevertheless, it is well known that speech production involves notable nonlinear processes [Kubin 95]. During the last few years there have been several attempts to turn towards nonlinear prediction [Tishby 90, Townshend 91, Wu 94, Kumar 97], but the results (with some exception) have not been very encouraging: the nonlinear models require a high computational effort compared to linear ones while the corresponding improvement is not so much. In other words, as long as nonlinear modeling does not significantly improve it is not worthwhile.

From our point view, the most relevant contribution to the field of nonlinear speech processing sets out the problem of speech analysis from the dynamical system theory [Kubin 95], trying to learn the *signal structure*. In this framework, the speech signal is seen as generated by a nonlinear dynamical system defined by a (low-dimensional) state-space vector and its evolution through a state space.

Unfortunately, the actual state-vector only can be inferred for quite simple systems, and as anyone can imagine, the dynamical system underlying the speech production process is very complex. Nevertheless, as established by the "embedding theorem" [Ott 93], it is possible to reconstruct a state space equivalent to the original one. Furthermore, a state-space vector formed by time-delayed samples of the observation (in our case, the speech samples) could be an appropriate choice:

$$\mathbf{s_n} = [s(n), s(n-T), \ldots, s(n-(d-1)T)]^t$$

where $s(n)$ is the speech signal, d is the dimension of the state-space vector, T is a time delay and t means transpose.

Finally, the reconstructed state-space vector dynamic, $\mathbf{s_{n+1}} = \mathrm{F}(\mathbf{s_n})$, can be learned through either local or global models, which in turn will be polynomial mappings, neural networks, etc.

Considering the reconstructed state-space vector $\mathbf{s_n}$ two questions naturally arise: What should be the embedding dimension of the (reconstructed) state-space vector, d ? And what should be the time delay, T ? Most of the researchers who have recently proposed nonlinear speech predictors have assumed $T = 1$ (following the linear case). Gernot Kubin [Kubin 95] was the first, up to our knowledge, to suggest that T should not be equal to 1 and to direct the attention of the speech community towards this "detail". Moreover, Bernhard and Kubin proposed a new method (computationally more efficient than the Fraser's one [Fraser 89]) to calculate both the dimension and the time delay [Bernhard 94].

Recently Abarbanel et al. [Abarbanel 98] reviewed the state of art concerning the techniques to deal with nonlinear deterministic systems. Following their exposition, the two first analysis steps intend to choose the time delay and the embedding dimension; however, in contrast to the already mentioned algorithms by Bernhard and Fraser, they suggest to determine them sequentially: first, the time delay ("independently" of the embedding dimension); and second, the embedding dimension.

In this paper we propose to apply the analysis techniques described in [Abarbanel 98] to the speech prediction problem, trying to learn the speech dynamics. Specifically, we use a Radial Basis Function (RBF) network [Haykin 99] to implement the nonlinear mapping that describes the evolution of the reconstructed state-space vector. Furthermore, we have conducted some experiments to gain insight into the sensibility of the prediction to both T and d .

II. Determining the time delay and the embedding dimension

It follows a brief summary of the methods to determine the time delay and the embedding dimension presented in [Abarbanel 98].

A. Average Mutual Information

When seeking the best value for T, the fundamental issue is to establish a right balance between a too small value (samples in the reconstructed state-vector exhibit a lot of common information) and a too large one (samples are independent). Abarbanel et al. suggest the following prescription: choose the value corresponding to the first minimum of the average mutual information I:

$$I(T) = \sum_{s(n),s(n+T)} P(s(n), s(n+T)) \log_2 \left[\frac{P(s(n), s(n+T))}{P(s(n))P(s(n+T))} \right]$$

where $P(\cdot)$ represents a probability which is estimated through a histogram.

B. False Nearest Neighbors

Now the issue is to determine the embedding dimension. For that purpose, Abarbanel et al. suggest the false nearest neighbors algorithm which is based on the following reasoning. For any point, we can ask whether its nearest neighbor is there due to the dynamics itself or is instead projected due to a too small reconstructed state-space vector dimension. Thus, the algorithm will compute the percentage of false nearest neighbors (those that disappear when the dimension is increased) for each of the candidate dimensions and will decide that the suitable dimension will be that for which the percentage of false nearest neighbors becomes zero (the dimension is then high enough).

III. RBF-based speech prediction

We use a RBF network to learn the dynamic of the reconstructed state-space vector. The RBF network is a single-layer network that computes the formula:

$$F(\mathbf{s}) = \sum_{k=0}^{M-1} c_k G(\|\mathbf{s} - \mathbf{t}_k\|)$$

where $\{G(\cdot)\}$ are RBF, $\{\mathbf{t}_k\}$ are the RBF centers, $\{c_k\}$ are the weights of the linear combination, and M is the number of RBF used. We use Gaussian RBF:

$$G(x) = \exp\left(-\frac{x^2}{\sigma^2} \right),$$

being σ its variance or width.

We have chosen the RBF network for this task for three main reasons: 1) it is a universal approximator; 2) the computational cost of its training is small compared to other types of networks; and 3) it yields a regularized solution to the prediction problem. This means that we seek a smooth solution, which offers good predictions in the regions where training data is not available. A compromise exits between smoothness and closeness to the data that is controlled through a regularization parameter, λ.

Given a training set composed of N pairs $(\mathbf{s}_n, s(n+1))$, we train the RBF network in two stages. First, the centers are obtained through a vector quantization algorithm, and the variance is computed as the maximum distance between centers. Second the output weights are determined as follows:

$$\mathbf{c} = \left(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{G}_0 \right)^{-1} \mathbf{G}^T \mathbf{s}$$

where

$$\mathbf{G} = \begin{bmatrix} G(\|\mathbf{s}_0 - \mathbf{t}_0\|) & G(\|\mathbf{s}_0 - \mathbf{t}_1\|) & \cdots & G(\|\mathbf{s}_0 - \mathbf{t}_{M-1}\|) \\ G(\|\mathbf{s}_1 - \mathbf{t}_0\|) & G(\|\mathbf{s}_1 - \mathbf{t}_2\|) & \cdots & G(\|\mathbf{s}_1 - \mathbf{t}_{M-1}\|) \\ \vdots & \vdots & \ddots & \vdots \\ G(\|\mathbf{s}_{N-1} - \mathbf{t}_0\|) & G(\|\mathbf{s}_{N-1} - \mathbf{t}_1\|) & \cdots & G(\|\mathbf{s}_{N-1} - \mathbf{t}_{M-1}\|) \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{M-1} \end{bmatrix},$$

$$\mathbf{G}_0 = \begin{bmatrix} G(\|\mathbf{t}_0 - \mathbf{t}_0\|) & G(\|\mathbf{t}_0 - \mathbf{t}_1\|) & \cdots & G(\|\mathbf{t}_0 - \mathbf{t}_{M-1}\|) \\ G(\|\mathbf{t}_1 - \mathbf{t}_0\|) & G(\|\mathbf{t}_1 - \mathbf{t}_2\|) & \cdots & G(\|\mathbf{t}_1 - \mathbf{t}_{M-1}\|) \\ \vdots & \vdots & \ddots & \vdots \\ G(\|\mathbf{t}_{M-1} - \mathbf{t}_0\|) & G(\|\mathbf{t}_{M-1} - \mathbf{t}_1\|) & \cdots & G(\|\mathbf{t}_{M-1} - \mathbf{t}_{M-1}\|) \end{bmatrix}$$

$$\text{and } \mathbf{s} = \begin{bmatrix} s(1) \\ s(2) \\ \vdots \\ s(N+1) \end{bmatrix}$$

IV. Experiments and Results

A. *Time delay and embedding dimension*

It is widely acknowledged [Kubin 95] that unvoiced sounds are properly modeled by linear methods, while voiced sounds demand a more elaborated nonlinear model. For this reason, our experiments have focused on sustained vowels. In particular, the experiments were made upon utterances of the five Spanish vowels (*a, e, i, o, u*) produced by one male and two female speakers, a total of fifteen voiced fragments of speech sampled at 8 kHz. For each one, the embedding dimension *d* and time delay *T* (in samples) were calculated as described above. Figure 1 illustrates both procedures for a particular case ("af1"). Table I shows the achieved results for all of the utternances.

Utterance	d	T
af1	7	3
af2	7	3
am1	7	3
ef1	5	5
ef2	6	5
em1	14	5
if1	6	6
if2	7	10
im1	7	7
of1	6	4
of2	6	4
om1	8	4
uf1	6	5
uf2	6	5
um1	6	6

Table I. Embedding dimension, *d*, and time delay, *T*, for each vowel sample (the 'm' stands for male and the 'f' for female).

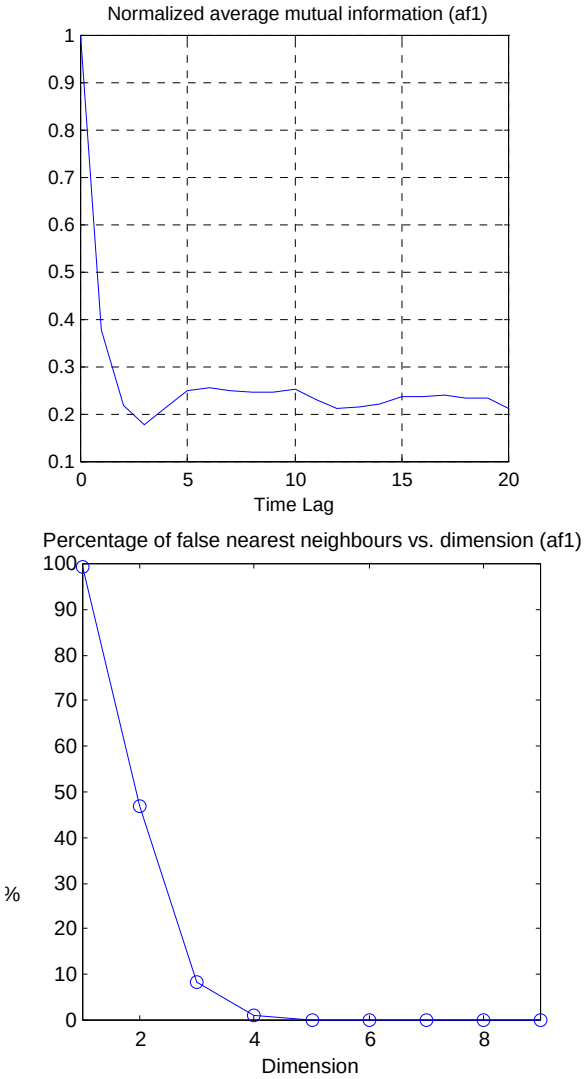

Figure 1. a) The first minimum of the average mutual information function is chosen as the Time Delay (T=3 in this case). b) Evolution of the percentage of false nearest neighbours as dimension is increased.

B. *Prediction experiments*

As previously mentioned, we use RBF networks to learn the state-space vector dynamic. In particular, the RBF networks, with $M = 80$ centers and $\lambda = 10^{-2}$, were trained over 300 samples in order to recursively predict 20 samples ahead; so the training window (300 samples of length) moves forward 20 samples for each 20 sample prediction (Figure 2 illustrates the procedure). Then the segmental signal-to-noise ratio *SNRseg* (a geometric mean of the SNR computed upon every 20 samples) was calculated over 25 frames (a period of 500 samples). Modification of the number of centers did not produce any significant changes in the *SNRseg* results. Also, an experimental exploration of

the values of the regularization parameter λ showed that 10^{-2} was convenient for all the voice samples.

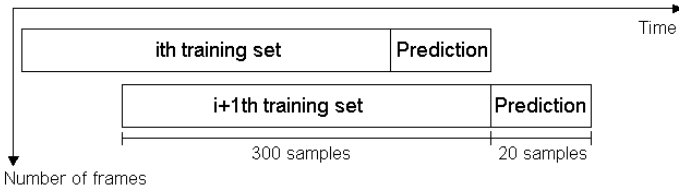

Figure 2. Training window moving for each 20 sample prediction

Our first prediction experiments provided poor results due to that the dimension of the state-vector and consequently the dimension of the centers was small. Specifically, the RBF network works better at a higher dimension because the matrix $(\mathbf{G}^T\mathbf{G}+\lambda\mathbf{G}_0)$, inverted during the training process, tends to be better conditioned as dimension grows (the probability of being a row a linear combination of others decreases). Therefore, we will use a dimension d' larger than that shown in Table I. In particular, we will use $d'=4d$.

For the last years, almost every attempt of nonlinear speech prediction had always considered $T=1$ so that the predicted sample was inferred from previous consecutive ones. Here a comparison between the conventional method ($T=1$) and the one proposed in this paper is presented. Henceforth, T_c and d_c will denote the time delay and embedding dimension in the conventional method (therefore, $T_c=1$); while T_s and d_s (with $d_s'=4d_s$) will denote those shown in Table I, i.e., those proposed in this paper.

We have compared both approaches in two ways: 1) considering similar computational costs (thus, using similar values for d_c and d_s'); and 2) making both predictors to observe the same amount of time memory (in this case $d_c=d_s'\cdot T_s$). Results in terms of SegSNR (in dB) are shown in Table II for all of the utterances. As can be deduced, the proposed approach provides much better results than those achieved by the conventional method for similar computation load (case I). While the results are comparable when both methods predict from the same time interval of the evolution of the state-vector (case II); nevertheless, it should be noticed that in this case the computational load of the conventional method is remarkably superior.

Finally, it is worthwhile noting that the results achieved for the utterance *em1* are especially poor. In our opinion, it should be due to that this utterance is not predictable 20 samples ahead.

Vowel	Utterance	Suggested Method (d_s',T_s)	Conventional Method (I) $(d_c=d_s',T_c)$	Conventional Method (II) $(d_c=d_s'T_s,T_c)$
a	af1	17.91	15.88	18.11
	af2	13.31	9.42	14.05
	am1	16.75	4.46	17.43
e	ef1	17.34	15.83	20.39
	ef2	14.27	13.68	14.66
	em1	1.75	13.86	3.47
i	if1	16.20	17.98	17.53
	if2	15.5	20.43	5.74
	im1	16.13	14.53	18.10
o	of1	23.45	22.34	24.86
	of2	22.11	21.29	23.05
	om1	23.19	14.28	23.77
u	uf1	26.38	25.53	27.03
	uf2	23.31	25.58	23.94
	um1	24.47	18.36	24.48
Mean		**18.14**	**18.44**	**16.9**
Standard deviation		**6.14**	**6.85**	**5.73**

Table II. SNRseg (dB) results for the proposed method (d_s',T_s) and the conventional one for both, similar computational cost $(d_c=d_s'T_s,T_c)$ and observed memory $(d_c=d_s',T_c)$.

C. *Sensibility of prediction to time delay*

Since the value of the embedding dimension is fixed by the RBF, we have studied the sensibility of the nonlinear prediction to the optimal values of T. This is a novel initiative that will allow us to know whether it could be worthwhile (without considering at this stage the computational cost) to choose an ad hoc time delay for each particular realization of a sound, or, on the contrary, a "mean" value would be acceptable.

We have performed a couple of experiments. First, we have selected one utterance, *am1*, and computed the prediction results for $d = d_s'$ and values of T between 2 and 11 (being $T_s = 3$). Figure 3 displays the achieved results. As shown, the best results are consistently achieved around $T = T_s = 3$, although a slightly better spurious result has been obtained for $T = 7$.

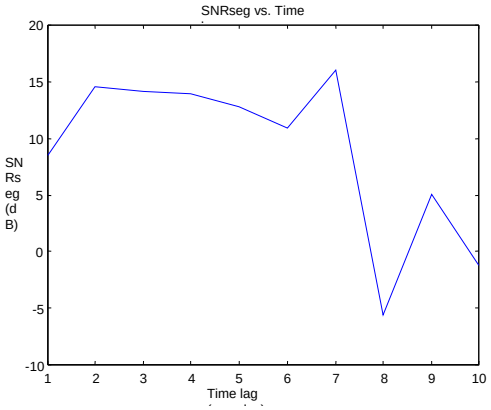

Figure 3. Prediction performance for utterance *am1* and several values of the time delay T.

Vowel	**a**	**e**	**i**	**o**	**u**	**All**
T	3	5	7	4	6	5

Table III. Time delays for each vowel and the global value.

Second, using $d = d_s'$, we have computed a general time delay for each vowel (extracted from the three samples of each vowel), hereafter T_{vowel}, and one global time delay (obtained from the whole of the vowel set), hereafter T_{global}. To be more precise, each set of signals corresponding to a same vowel was unified into a single signal by simply juxtaposing the utterances of each speaker, and for the new five signals, five new values of *T* were calculated, giving as result a time delay *T* for each vowel, T_{vowel}. Moreover, all the utterances were put together in the same way and what could be considered a global value of T_{global} was obtained. Once computed this values of *T*, shown in Table III, our experiment consisted in, obtaining the prediction error for these general values of *T*.

Vowel	Utterance	T_s	T_{vowel}	T_{global}
a	af1	17.91	17.97	16.57
	af2	13.31	13.30	8.93
	am1	16.75	16.71	15.87
e	ef1	17.34	20.28	17.36
	ef2	14.27	14.20	14.29
	em1	1.75	18.30	1.75
i	if1	16.20	16.11	18.27
	if2	15.5	18.89	14.34
	im1	16.13	13.93	17.12
o	of1	23.45	24.58	24.35
	of2	22.11	23.71	22.14
	om1	23.19	23.91	21.10
u	uf1	26.38	25.46	26.37
	uf2	23.31	24.57	23.34
	um1	24.47	24.62	25.48
Mean		**18.14**	**19.77**	**17.82**
Standard deviation		**6.14**	**4.40**	**6.56**

Table IV. Segmental signal to noise ratios in dB for the recursive prediction of 500 samples in steps of 20, using three values of *T*.

Using the three values of T obtained (T_s, T_{vowel}, and T_{global}) RBF networks were trained over 300 samples, in order to forecast recursively 20 samples, repeating the experiment again over 500 samples. The results are shown in Table IV.

As it can be inferred from results shown in Table IV, the achieved results are not sensitive to the specific value computed for each utterance, as long as a close value is used (the same conclusion can be drawn from Figure 3).

V. CONCLUSIONS AND FURTHER WORK

We have presented a speech prediction system which offers similar performances to that of the conventional method but with a much smaller computational cost. The new approach consists in predicting from a vector of non-consecutive samples, being *T* the distance between samples. The dimension of the vector *d* and the distance *T* have been estimated following the algorithms described by Abarbanel et al. [Abarbanel 98]

Furthermore, we have studied the sensibility of the prediction performance to the value of *T*, concluding that any value of *T* in the neighbour of the optimal one can be used without a relevant loss of performance.

The type of predictor we have used, RBF networks, has made us to work with higher embedding dimensions than necessary. We are currently working on the same ideas using other types of predictors in order to circumvent the constraints imposed by the RBF networks.

Finally, the next step in our research will focus on determining a prediction horizon by studying the Lyapunov exponents.

VI. REFERENCES

[Abarbanel 98] H.D.I. Abarbanel, T.W. Frison and L.S. Tsimring: Obtaining Order in a World of Chaos; IEEE Signal Processing Magazine, vol. 15, no. 3, pp. 49-65; 1998.

[Bernhard 94] H.-P. Berhhard and G. Kubin: "A Fast Mutual Information Calculation Algorithm"; in Proc. VII European Signal Processing Conference, EUSIPCO-94, pp. 50-53; Edinburgh, Scotland, U.K.; 1994.

[Fraser 89] A.M. Fraser: Information and Entropy in Strange Attractors"; IEEE Trans. on Information Theory, vol. 33, no. 2, pp. 245-262; 1989.

[Haykin 99] S. Haykin, *"Neural Networks: A Comprehensive Foundation"*, Second Edition; Upper Saddle River, NJ: Prenctice Hall, 1999.

[Kubin 95] G. Kubin: "Nonlinear Speech Processing"; in *Speech Coding and Synthesis*, pp. 557-610; W.B. Kleijn, K.K. Paliwal, Ed.; Amsterdam, The Nethelands: Elsevier Science; 1995.

[Kumar 97] A. Kumar and A. Gersho: "LD-CELP Speech Coding with Nonlinear Prediction"; IEEE Signal Processing Letters; vol. 4, no. 2, pp. 89-91; 1997.

[Ott 93] E. Ott: *"Chaos in Dynamical Systems"*; Cambridge: Cambridge University Press, 1993.

[Tishby 90] N. Tishby: "A Dynamical System Approach to Speech Processing"; Proc. ICASSP-90, vol. I, pp. 365-368; Albuquerque, New Mexico; 1990.

[Townshend 91] B. Townshend: "Nonlinear Prediction of Speech"; Proc. ICASSP-91, vol. I, pp. 425-428; Toronto, Canada; 1991.

[Wu 94] L. Wu, M. Niranjan and F. Fallside: "Fully Vector-Quantized Neural Network-Based Code-Excited Nonlinear Predictive Speech Coding"; IEEE Trans. on Speech and Audio Processing, vol. 2, no. 4, pp. 482-489; 1994.